Request For Comments: 886

Proposed Standard for Message Header Munging

Thu Dec 15 03:37:52 1983

Marshall T. Rose

Department of Information and Computer Science
University of California, Irvine
Irvine, CA 92717

MRose.UCI@Rand-Relay

This memo proposes a standard for the ARPA Internet community. If this proposal is adopted, hosts on the ARPA Internet that do message translation would be expected to adopt and implement this standard.

Introduction

This memo describes the rules that are to be used when mail is transformed from one standard format to another. The scope of this memo is limited to text messages (computer network mail, or electronic mail) that traverse the ARPA Internet. This memo is not presented as a replacement or amendment for the "Standard for the Format of ARPA Internet Text Messages", RFC822. Rather, this memo focuses on a particular aspect of mail, and provides a conceptual and practical basis for implementors of transport agents and user agents which support message munging.

Although this memo has been specifically prepared for use with the 822 standard, an understanding of the 822 standard is not required to make use of this memo. The remainder of this section reminds the reader of some key concepts presented in the 822 standard, and how they relate to the perspective of this memo.

Messages are viewed as consisting of an envelope and contents. The envelope is manipulated solely by transport agents, and contains the information required by the transport agents to deliver the message to its recipients. Although this memo does not address itself directly to the envelope, we shall see that some of the rules discussed later are applicable to the envelope.

The contents of the message consists of a rigorously structured part, known as the headers, followed by a freely formated part, called the body. The message body is completely uninteresting to us. Our emphasis is strictly on the headers of the message. Each header in the message consists of a field, its value, and a terminating end-of-line sequence. The 822 standard discusses, among other things, continuation lines, the syntax that is used to distinguish between fields and values, and the syntax and semantics of the values of various fields. For our part, we shall concern ourselves only with the notion that the headers section consists of one or more headers, which are divided into one or more field/value pairs.

The term "message munging" refers to the actions taken by a transport or user agent to transform the contents of a message from conformance with one standard format to another. The 822 standard refers to this as "Network-Specific Transformation". Other phrases might be "header munging" or "mail filtering". Regardless of the term used, the key notion is that this action transforms a message from its current format (the source message) to the structure required by the target standard. A "munging agent", for the purposes of this memo, is an entity which performs message munging. A munging agent may be part of either a transport or user agent.

Background

As more networks connect into the ARPA Internet community, their users will exchange computer mail messages with other Internet hosts. Although the 822 standard must be strictly adhered to for mail that traverses the ARPA Internet, other networks might not internally adopt this standard. It is nevertheless desirable to permit mail to flow between hosts which internally conform to the standard and those which do not. The 822 standard is very clear to indicate that:

> "This standard is NOT intended to dictate the internal formats used by sites, the specific message system features that they are expected to support, or any of the characteristics of user interface programs that create or read messages."

This plainly states that even hosts within the ARPA Internet, may opt to use a different standard than 822 for their internal use, but they are expressly required to use the 822 standard when transferring mail to other hosts in the ARPA Internet. As such, it is not difficult to imagine message munging becoming a common activity among transport and user agents.

There are other reasons why message munging may become a widespread practice. An example from CSnet will serve here. The CSnet relays provide authorized access for mail services to the ARPA Internet for the CSnet phonenet sites. CSnet sites are not registered with the NIC, and hence are often absent from the host tables of ARPA Internet sites. As a result, addresses for mailboxes on CSnet phonenet sites are unknown to ARPA Internet sites. From an ARPA Internet site, it would be impossible to send messages to these addresses, since the local transport agent has no handle on the destination hosts of the phonenet mailboxes. Obviously, even replying to a such a message is simply not possible. To solve this problem, the transport agents on the CSnet relays perform message munging on mail destined for the ARPA Internet. Phonenet addresses of the form "mbox@host" are transformed to "mbox.host@relay", where "relay" is the ARPA Internet host name of the relay performing the transformation. Other addresses are left alone. Agents throughout the ARPA Internet are now able to process these addresses, since the host-part is a known ARPA Internet host.

The source-routing solution to this problem will hopefully be replaced by domain handling when domains are implemented in the ARPA Internet. When this is the case, phonenet addresses of the form "mbox@host" will become "mbox@host.CSNET". Despite this change, (which cannot help but be for the better, as the use of source-routing leads to a plethora of problems), message munging will still occur as it will most likely be necessary to add domain names during message transmission (see section 6.2.2 of the 822

standard).

For an alternate reason, consider that it is not unlikely for users to wish to transform mail from their archives which conforms to an older standard to the current standard. There could be many reasons for this, although a common one would be that a user wishes to re-introduce the message into the transport system. Although the aged message was perfectly valid when it was composed (e.g., in the days of the 733 standard), it might no longer conform to the current standard (i.e., 822). In this case, a user agent would have to perform message munging, in order to make the message acceptable to the local transport agent.

To summarize, even under the most "homogeneous" of environments, message munging will still be required on the part of transport and user agents, under certain conditions.

Section 3.4.10 of the 822 standard briefly discusses the topic of "Network-Specific Transformations". In short, the 822 standard envisions a message traversing net-A to reach net-B as going through three phases:

- o Transformation
  The message is made to conform to net-A's standards

- o Transformation Reversal
  Net-A's idiosyncrasies are removed and the message now conforms to the 822 standard

- o Transformation
  The message is made to conform to net-B's standards

This memo concerns itself solely with this section of the 822 standard. The 822 standard presents end-of-line sequences as an example of an area where transformation might occur. Although this is a valid concern, our emphasis deals with constructs of higher semantics: fields and structured field values.

Scope

This memo does not specify the particular transformation rules that should be used when munging a message from one standard to another.

Rather, this memo attempts to make clear the policies that are to be followed when implementing any munging agent for the ARPA Internet. The derivation of the formulas specific to message munging between two given standards is left to the implementors of such munging systems or to the writers of future RFCs. As such, this memo can be considered to present the philosophy and conceptual basis of message munging in the ARPA Internet.

> NOTE: It is critical that this position be understood. The actual policies used by domain-specific munging agents is completely beyond the scope of this memo.

For ease of explanation, some of the examples in this memo use message munging between the ARPA Internet and the USENET distribution network as an example. This memo should NOT be considered to specify how this particular munging activity should take place. Instead, this context has been chosen for its familiarity and simplicity.

Message Decomposition

A munging agent concerns itself with transforming a message in conformance with a source standard to a message in conformance with a target standard. This transformation occurs at various levels. Four of these are presented here.

o Field Transformation

> For two standards, some fields may convey identical semantics but have different names. As standards progress, for example, the names of fields may change, but the presence of those fields and their contents continue to have the same meaning. For example, prior to 822 standard, some mailers considered the Remailed- prefix to have semantics equivalent to the 822 standard's Resent- prefix. In this circumstance, one aspect of message munging would be to simply substitute the field names.

o Value Transformation

> The value of certain fields may be viewed as containing

structured components. The syntax and semantics of these components may differ significantly between two formats. In this circumstance, one aspect of message munging would be to transform components from one representation to another.

o Field/Value Combination

The semantics of a given header in a particular standard may not be directly expressed using a single header from another standard. In this circumstance, one aspect of message munging would be to map the field/value of a header in the source message to any number of headers in the target message (or vice-versa). As expected, further complication could result by performing value transformation in addition to one-to-many or many-to-one field transformation.

o Header Ordering

Some standards may require that fields appear in a particular order in the headers part of the message. Others make no requirements as to the order in which the fields appear. In this circumstance, one aspect of message munging from the latter to the former standard would be to capture the essential information from the source message in order to construct the first field of the target message. As expected, further complication could result by requiring several field/values be consulted in the source message before sufficient context is present to construct the first field of the target message.

Canonical Forms

Fundamental to the activity of transformation is the notion of a canonical form. For a given message standard, each field and structured field value may be thought of as an object with a particular semantics that is representable by one or more strings. That is, each of these strings has an identical semantics, as they all refer to the same object. For example, in terms of the 822 standard, the two strings

    The Protocol Police <NetSer@UCI>

    NetSer@UCI

are semantically equivalent. For the purposes of this memo, a fully-qualified canonical form of an object is thought of as the simplest string that represents the full and complete meaning of an object. The meaning of "simple" is, of course, open to interpretation. In some cases, "simplest" may mean "shortest". In other cases, a longer, but unabbreviated string may be "simpler" than a shorter string. Regardless of this, a canonical form is a representation of an object. This representation contains the smallest amount of information required to fully describe the meaning of the object.

It is not difficult to determine what a canonical form should describe for different objects. In terms of the 822 standard, the following should be considered as minimal definitions of canonical forms:

object	specifies	contains
field	field-name	name
address	mailbox	local-part
		domain-part
date	date-time	date-part
		time-part

In terms of USENET, the following might be considered as minimal definitions of canonical forms:

object	specifies	contains
field	field-name	name
address	mailbox	user
		route
date	date-time	date-part
		time-part

NOTE: This memo clearly has no authority to specify the

minimal canonical forms for USENET. The above table is listed solely for the benefit of the examples which follow.

Conceptually, transformation of fields and structured field values occurs between canonical forms. That is, to transform an address, one reduces the string representing the object to its canonical form, to capture the essence of its meaning, and this form is then transformed, somehow, to the equivalent canonical form for the target standard. This target canonical form can later be output using a string representation.

NOTE: This memo does not require that canonical forms be represented or otherwise implemented as strings. Nor does this standard require that strings be used during the transformation process. Thinking of a canonical form as a string is a convenient formalism only, not an implementational requirement.

Transformation Rules

All of the forms of message *decomposition* discussed above may now be viewed as transformation between canonical forms. Hence, it now becomes necessary to consider how canonical forms should be manipulated during transformation. That is, what rules are to be followed when constructing an equivalent canonical form? There are several guidelines that must be followed, that we will characterize in the following fashion:

o Preservation of information

All attempts must be made to preserve all information contained in the original canonical form. This information can be highly useful to the recipients of munged messages. Obviously, for two widely-differing formats, this may not be possible. For example, some standards may not have a group addressing notation such as the one present in the 822 standard, e.g., the notation

```
List: Support@UCI, ZOTnet@UCI;
```

might not be permitted. If one were to consider membership in a group as part of an address' canonical form, then this portion of the canonical form could not be transformed to the other standard.

The 822 standard supports a liberal commenting convention which might prove quite useful in preserving information. Implementors may wish to consider capturing the original information in commentary text. For example, if the USENET address

```
mark@cbosgd.UUCP (Mark Horton)
```

had the USENET canonical of

```
 user:  mark
route:  ucbvax!ihnp4!cbosgd
```

and if the corresponding 822 canonical was

```
 local-part:  ucbvax!ihnp4!cbosgd!mark
domain-part:  USENET.UCI
```

then it would not be unreasonable for an implementation to output this canonical form as

```
"mark@cbosgd.UUCP" <ucbvax!ihnp4!cbosgd!mark@USENET.UCI>
```

NOTE: Implementors should exercise extreme caution in using a policy such as this. Information placed between commentary delimiters must still conform to the target standard at the syntactic level.

Note however that in the above example, the commentary information "(Mark Horton)" was discarded. This practice is strongly discouraged. Although the canonical form for an object does not rely on commentary information as a necessary part, implementors are encouraged to preserve this information whenever possible.

Finally, preservation of information requires preservation of case at all costs. Only under the most restrictive of circumstances should an implementation change the case of the strings output for a canonical form.

o Re-Formatting

Ideally, the target message should have the exact horizontal and vertical padding as the source message. Because a string representing the source canonical form of an object may not be of the same length as the string representing the target canonical form, the number of characters on each physical and logical line in the headers may be different.

The 822 standard supports a header folding convention which permits long field/value pairs to be represented on more than one physical line. When a new canonical form is output to the target message, it is possible that the resulting field/value pair may be longer than the number of characters that antiquated display devices can present on a single line. The 822 standard suggests 65 or 72 characters-per-line as a metric for this limitation. Although not required, message munging agents may re-fold headers if (and only if) this limitation is exceeded. Note however that under no circumstances should a header be re-folded if it was not munged. Refolding without munging may occur on behalf of some transport or user agent, but it may not occur on behalf of a munging agent. Put more simply, this memo does not authorize or forbid such activity, although it does discourage it.

o Error Recovery

The preceding discussion has made been under the assumption that the objects composing the field/value pairs of the source message have conformed to the source standard. It is an unfortunate reality that this may not be the case. In fact,

for those standards which are poorly specified (if at all), determining that an object is improperly constructed might be quite difficult. In addition, it is possible, though hopefully extremely improbable that a target canonical form does not exist for a particular source canonical form. In these cases, munging agents must be able to recover.

At this point, we introduce two extension fields for the 822 standard. As such, these fields are hereby designated as "reserved" and may not be used for other purposes. These fields are:

```
Illegal-Field
Illegal-Object
```

The syntax of these fields is as follows:

```
munge-field =
     "Illegal-Field"    ":" *text
   / "Illegal-Object"   ":" *text

munge-object =
     <a "field-name", the exact field-names which are
      valid will be presented later>
```

The semantics of these fields are as follows:

An Illegal-Field header should be introduced when a header-line which does not conform to the source standard is found in the source message. Illegal-Field should be used only when a header-line is so poorly formed as to prevent recognition of the field in the header-line. For example, if the line lacks a colon, or has a poorly formed field-name, then it should not be output to the target message and a new header-line should be introduced in its place. This header-line has Illegal-Field as its field and contains the offending line as its value. Illegal-Field should not be used if the field can be identified, but the value is poorly formed.

An Illegal-Object header should be introduced when an object in the source message can not be parsed into a canonical form, or if the canonical form it represents has no corresponding target canonical form. The offending object should not be output to the target message in the header-line in which it occurs. If the header-line now contains no objects, then the header-line should not be output to the target message as well. Then, an Illegal-Object field should be introduced into the target message. The value of this Illegal-Object field should at the very minimum contain the name of the field that contained the object, the object in question, and an

explanation as to why the object was illegal. Alternately, the value of this Illegal-Object field should consist of the entire header-line (field and value) that contained the object in question along with an explanation as to why the object was illegal.

NOTE: In the circumstance where multiple objects exist in a single header-line in the source message, and one of those objects is determined to be illegal, the actual policy used in determining how much information can be considered to be "uncorrupted" is left to the implementors. Munging agents which use sophisticated parsers may attempt to recover in mid-stream (so to speak) and continue parsing objects on the header-line. Other agents may wish to continue recover with the next header-line in the source message. Regardless of the policy used, the agent must present the contents of the entire header-line in the associated Illegal-Object header.

Implementations should not take extraordinary measures to perform syntax/semantics checking of the source message -- only those fields which must be examined should be rigorously checked. This memo strongly discourages any additional examination. It is not the intention of this memo to suggest that composing agents should produce messages which do not conform to the source standard. A composing agent should not expect a munging agent to enforce adherence to the source standard.

o Introduction of Information

Munging agents are authorized to introduce a "Received" header into the target message when a message is transformed.

NOTE: Adding a "Received" header is entirely optional. This memo strongly recommends that this header be introduced whenever some munging (translation of addresses and/or dates) occurs.

NOTE: Although this memo does not specify the position that the introduced header should have in relation to the other fields in the target message, it is strongly recommended that the introduced header be grouped with the other "Received" headers, at the very beginning of the message.

When introducing a "Received" field, three phrases, which are normally optional in such a field, should be specified by the munging agent. These are:

```
"from" domain       # the source domain
"by"   domain       # the target domain
"with" protocol     # the munging agent's host
```

For example, suppose we have a munging agent on the UCI host, and that this agent services a USENET/ARPA boundary. When the UCI host gets a message from the USENET domain for the ARPA domain, the following happens: First, the UCI mailsystem would prepend the header:

```
Received: from nma by UCI with UUCP; 15 Dec 83 03:53:00 PST
```

Second, the munging agent, when transforming the message, would prepend the header:

```
Received: from USENET by ARPA with UCI; 15 Dec 03:54:00 PST
```

Finally, the UCI mailsystem would then deliver the message to the appropriate ARPA mailsystem, which in turn would prepend the header:

```
Received: from UCI by ISIF with SMTP; 15 Dec 83 03:55:00 PST
```

This example might be a bit clearer if the domains were qualified a bit more. The first three lines of the message could look like this:

```
Received: from UCI.ARPA by ISIF.ARPA; 15 Dec 83 03:55:00 PST
Received: from USENET by ARPA with UCI; 15 Dec 03:54:00 PST
Received: from nma.USENET by UCI.USENET; 15 Dec 83 03:53:00 PST
```

The key point to notice is that the munging agent used the "from" and "by" clauses to denote the domain boundary that was crossed, and used the "with" clause to denote itself. Since the agent is munging the message according to some set of transformation rules, it is actually using a "mail protocol", and as such is justified in identifying itself in the "with" clause.

Objects of Interest

At present, only three types of objects are of interest: fields, addresses, and dates. In the context of the 822 standard, header-lines containing the following fields are to be viewed as appropriate for transformation:

Address Fields:	From, Sender, Reply-To, To, cc, Bcc, and any of these fields with the Resent- prefix
Date Fields:	Date, Resent-Date

Hence the definition of munge-object, in 822 terms, is:

```
munge-object =
      "From"
    / "Sender"
    / "Reply-To"
    / "To"
    / "cc"
    / "Bcc"
    / "Resent-From"
    / "Resent-Sender"
    / "Resent-Reply-To"
    / "Resent-To"
    / "Resent-cc"
    / "Resent-Bcc"
    / "Date"
    / "Resent-Date"
```

NOTE: The list of munge-objects is extensible. For the purposes of this memo, the above fields are defined as the MINIMUM list of munge-objects for the 822 standard. Implementors are encouraged to introduce other fields to the list of munge-objects as their munging agents require. These additions should also be registered with the revisions of this memo as they gain popularity.

For the purposes of the remainder of this memo, an address header-line is defined as any header-line in the source message whose field component is one of the fields listed above as an address field. Further, a date header-line is defined as any header-line in the source message whose field component is one of the fields listed above as an date field.

If address munging is performed, then all addresses contained in all address header-lines must be munged. It is expressly forbidden to perform address munging on the source message and without performing address munging on every address header-line. Further, it is expressly forbidden to munge some, but not all, of the addresses in any address header-line. All addresses in all of the

message's address header-lines must be address munged. If address munging is not performed, then these header-lines need not be considered for munging.

A similar requirement is made for date munging. If date munging is performed, then all instances of header-lines whose field is Date or Resent-Date must be fully date munged.

> NOTE: Certain fields are to be excluding from munging of any sort, all munging agents must preserve their contents exactly. At present, there is one such field: "Received". This contents of this field should ALWAYS be preserved for trace and debugging purposes.

Bibliography

D.H. Crocker, "Standard for the Format of ARPA Internet Text Messages", RFC 822, (August, 1982).

M.R. Horton, "Standard for the Interchange of USENET Messages", RFC 850, (June, 1983).

M.T. Rose, "Achieving Interoperability between two Domains -- Connecting the ZOTnet and UUCP Computer Mail Networks", Technical Report Number 201, Department of Information and Computer Science, University of California, Irvine, (January, 1983).

P.V. Mockapetris, "Domain Names -- Concepts and Facilities", RFC 882, (November, 1983).

Appendices

Minimal Canonical Forms

This memo defines the minimal canonical forms for the 822 standard. Implementations may wish to augment these forms with additional information that may be present in the source message. An earlier example suggested that group membership might be part of an address' canonical form. Further, since the 822 standard permits routes to be specified in addresses, e.g.,

Fred Rated <@ISI-Troll.ARPA,@UCI-750a.UCI: FRated@UCI-750b>

Perhaps they too should be considered part of the 822 address' canonical form?

This memo makes no such requirement, if implementations wish to make use of this additional information, then they are free to do so. This practice is neither encouraged nor discouraged. In short the spirit of this memo is to require those minimal components required by the 822 standard, nothing more.